**Proposal for Secure Self-Destructing Notes Application**

**Project Title:** FadeNote: Secure Messages with a Self-Destruct Timer

A web app called FadeNote lets people make, handle, and safely share notes that disappear after a certain amount of time. This project was inspired by PrivNote.com and wants to make it easy for people to make and read notes while also ensuring strong security to keep user data safe.

**Application Workflow:**

1. **Access Website**: Users can access the website without needing to log in or sign up.
2. **Create Note**: Upon visiting the site, users are presented with the option to create a new note.
3. **Note Options**:
   * Users can set options for the note to self-destruct:
     + After being read once.
     + After one hour of being read.
   * Optionally, users can add a password for added security.
4. **Encryption**: Every note is encrypted and saved in the database securely.
5. **Share Notes**: Once created, notes can be shared through a generated URL.
6. **Read Notes**: Recipients can read the note without creating an account. The note will be deleted based on the chosen self-destruction criteria.

**Technology Stack:**

* **Backend**: Node.js with Express.js
* **Frontend**: EJS, Bootstrap, jQuery
* **Database**: MongoDB for secure and efficient data storage.

**Key Features:**

* **No User Accounts**: Simplifies the user experience by not requiring login or registration.
* **Self-Destruct Options**: Notes can self-destruct after being read or within a specified time.
* **Encryption**: All notes are encrypted before storage, ensuring secure data transmission and storage.
* **User Experience**: Streamlined note creation and sharing without complicated workflows.

**Milestones:**

1. **User Interaction**: Design an intuitive interface for note creation and reading without user registration.
2. **Note Creation & Encryption**: Implement note creation, encryption, and secure storage.
3. **Self-Destruct Mechanism**: Develop options for self-destruction based on user-defined parameters.
4. **Sharing & Security**: Implement secure sharing through unique URLs and optional password protection.
5. **Testing & Launch**: Comprehensive testing to ensure functionality and security before launching.

**Technical Workflow:**

A screenshot of a computer screen

Description automatically generated

**Conclusion:**

This plan describes how to make TempNote, a safe place to store notes that delete themselves after a certain amount of time. By putting the user experience first and putting in place strict security measures, TempNote will give users a reliable, user-centered app for keeping private information safe.

**Future Enhancements:**

* **Mobile App Development**: Develop native apps for iOS and Android.
* **Browser Extensions**: Enable note creation and saving directly from web browsers.
* **Multilingual Support**: Expand accessibility by supporting multiple languages.

**References:**

1. Node.js Official Documentation: [Node.js Documentation](https://nodejs.org/en/docs/)
2. MongoDB Official Documentation: [MongoDB Documentation](https://www.mongodb.com/docs/)
3. Message Encryption in Node.js: [Encryption Guide](https://habtesoft.medium.com/encrypt-data-on-nodejs-e0a45c67c772)
4. Secure Coding Practices: [OWASP Guide](https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/)